

de All my codes where : ↗ 31
20240575 : ID

Sheet 2

Q1) #include <stdio.h>

```
int main() {  
    int n, max, min, sum = 0;  
    scanf("%d", &n);  
    nums[n];  
    scanf("%d", &nums[0]);  
    max = nums[0];  
    min = nums[0];  
    sum += nums[0];
```

```
    for (int i=0 ; i<n ; i++) {  
        if (max < nums[i]) max = nums[i];  
        if (min > nums[i]) min = nums[i];  
        sum += nums[i];  
    }
```

```
    printf("The Average: %.2f\n The Minimum: %.d\n"  
          "The Maximum: %.d", (float)sum/n, min, max);
```

}

Q2) #include <stdio.h>

```
int main() {  
    int n, isascending = 1;  
    scanf("%d", &n);  
    int nums[n];  
    scanf("%d", &nums[0]);
```

continue in Page 2

Q2 continued

```
For (int i=1; i<n; i++) {  
    scanf("%d", &num[i]);  
    if (num[i] < num[i-1]) isAscending = 0;  
}  
if (isAscending) printf("The array is ascending");  
else printf("The array is not ascending");
```

Q3) #include <stdio.h>

```
int main() {  
    int n, sumEven = 0, sumOdd = 0;  
    scanf("%d", &n);  
    int num[n];  
    for (int i=0; i<n; i++) {  
        scanf("%d", &num[i]);  
        if (num[i] % 2 == 0) sumEven += num[i];  
        else sumOdd += num[i];  
    }  
    printf("The even sum = %d \n The odd sum = %d  
          , sumEven, sumOdd);
```

Q4) #include <stdio.h>

```
int main() {  
    int n; int same = 1, inc = 1, dec = 1, incInc = 0,  
        incDec = 0, incDecDec = 1;  
    scanf("%d", &n);  
    int num[n];  
    scanf("%d", &num[0]);
```

continued in Page 3

Q4 continued

```
ForC int i=1, i<n ; i++) {  
    ScanF ("%d", &nums[i]);  
    IF( nums[i] > nums[i-1]) {  
        same = 0;  
        dec = 0; }  
    IF( nums[i] < nums[i-1]) {  
        same = 0;  
        inc = 0; }  
IF( dec  
    IF( inc && !same)  
        incinc = 1;  
    IF( incinc && !inc)  
        incdec = 1;  
    IF( incdec && num[i] > num[i-1])  
        incdecdec = 0;  
    }  
}
```

```
IF(sam)  
    printf(" not changing");  
else IF(dec)  
    printf("decreasing");  
else IF(inc)  
    printf("increasing");  
else IF(incdec && incdecdec)  
    printf(" increasing then decreasing");  
else  
    printf(" no pattern");  
}
```

Q5) #include <stdio.h>

```

int main() {
    int rows = 3, cols = 4;
    int num, Found = 0;
    int nums[3][4];
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 4; j++) {
            scanf("%d", &nums[i][j]);
        }
    }
    printf("What number to search for? ");
    scanf("%d", &num);
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 4; j++) {
            if (num == nums[i][j]) {
                printf("Your number is at row %d, and
column %d", i + 1, j + 1);
                Found = 1;
            }
        }
    }
    if (!Found) printf("number not found");
}

```

Q6) #include <stdio.h>

```

int main() {
    int rows, cols, sum = 0, maxTotal, maxRow;
    printf("how many rows and cols? ");
    scanf("%d%d", &rows, &cols);
    int nums[rows][cols];

```

continued in Page 5

Q6 continued

```
ForC int i=0; i<rows ; i++) {  
    sum=0;  
    For Cint j=0 ; j<cols ; j++) {  
        scanf("%d", &num[i][j]);  
        sum += num[i][j];  
    }  
    if (i) {  
        maxtotal = (sum > maxtotal) ? sum : maxtotal;  
        maxrow = (sum > maxtotal) ? i+1 : maxrow;  
    }  
    else { maxtotal = sum; maxrow = i+1; }  
    printf(" Row %d has maximum sum of elements",  
           maxrow, maxtotal);  
}
```

Q7) #include <stdio.h>

```
int main() {  
    int n;  
    scanf("%d", &n);  
    int num[n][n];  
    ForC int i=0 ; i<n ; i++) {  
        ForC int j=0 ; j<n ; j++) {  
            if (i==j) printf("%2d\t", 0);  
            else if (i<j) printf("%2d\t", 1);  
            else if (i>j) printf("%2d\t", -1);  
        }  
        printf("\n");  
    }  
}
```

Q8) #include <stdio.h>

```

int main() {
    int n;
    scanf("%d", &n);
    int num[n][n];
    for (int i=0; i<n; i++) {
        for (int j=0; j<=i; j++) {
            if (i==j || j==0 || j==n)
                num[i][j] = 1;
            else
                num[i][j] = num[i-1][j] + num[i-1][j-1];
            printf("%d\t", num[i][j]);
        }
        printf("\n");
    }
}

```

Q9) #include <stdio.h>

```

int reverse(int num[], int revnum[], int n) {
    for (int i=0; i<n; i++)
        revnum[n-1-i] = num[i];
}

int main() {
    int n;
    scanf("%d", &n);
    int num[n], revnum[n];
    for (int i=0; i<n; i++) scanf("%d", &num[i]);
    reverse(num, revnum, n);
    printf("The original array:\n");
    for (int i=0; i<n; i++) printf("%d", num[i]);
    printf("\n");
    printf("The reversed array:\n");
    for (int i=0; i<n; i++) printf("%d", revnum[i]);
}

```

Q10) #include <stdio.h>

```
int AvgGrade( float stds[], [6], int n ) {  
    for( int i=0; i<n; i++ ) {  
        stds[i][5] = 0;  
        for( int j=0; j<5; j++ ) {  
            scanf( "%f", &stds[i][j] );  
            stds[i][5] += stds[i][j] / 5.0;  
        } } }  
int main() {  
    int n; scanf( "%d", &n );  
    float stds[n][6];  
    AvgGrade( stds, n );  
    for( int i=0; i<n; i++ ) {  
        printf( "The average grade for student %d is %.2f  
                , i+1, stds[i][5] );  
    } }
```

Q11) #include <stdio.h>

```
int ISSParse( int rows, int cols, int nums[ ][cols] ) {  
    int count = 0;  
    for( int i=0; i<rows, i++ ) {  
        for( int j=0; j<cols, j++ ) {  
            if( !nums[i][j] ) count++;  
        } }  
    if( count > rows * cols / 2 ) return 1;  
    else return 0;  
}
```

continued in Page 8

Q11 continued

```
int main() {
    int rows, cols; scanf("%d%d", &rows, &cols);
    int nums[rows][cols];
    for (int i=0; i<rows; i++) {
        for (int j=0; j<cols; j++) {
            scanf("%d", &nums[i][j]);
        }
    }
    if (ISSparse(rows, cols, nums))
        printf("Sparse");
    else
        printf("Not Sparse");
}
```

Q12) #include <stdio.h>

```
int ISSymm(int n, int nums[][n]) {
    int sym=1;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            scanf("%d", &nums[i][j]);
            if (i>j && nums[i][j] != nums[j][i])
                sym=0;
        }
    }
    return sym;
}

int main() {
    int n; scanf("%d", &n); int nums[n][n];
    if (ISSymm(n, nums)) printf("Symmetric");
    else printf("Not Symmetric");
}
```

}

(8)

Q13) ~~#include <stdio.h>~~
~~const float Pi = 3.14;~~
 float ~~Acircle~~ Acircle(float r){
 return ~~Pi~~ * r * r;
 }
 float cFcircle (float r){
 return 2 * Pi * r;
 }
 int main(){
 float r;
 scanf("%f", &r);
 printf("The circumference = %f\n The Area = %f
 , cFcircle(r), Acircle(r));
 }

Q14) ~~# include <stdio.h>~~
 int calc(int nums[], int n, int *max, int *min, float *avg)
 { for (int i=0; i<n; i++) {
 if (i){
 *max = (nums[i] > *max) ? nums[i] : *max;
 *min = (nums[i] < *min) ? nums[i] : *min;
 }
 else {
 *max = nums[i];
 *min = nums[i];
 }
 *avg += (float) nums[i] / n;
 } }

continued in Page 10

Q14) continued

```
int main(){
    int n, max, min;
    float avg=0;
    scanf("%d%n");
    int nums[n];
    for (int i=0; i<n; i++)
        scanf("%d", &nums[i]);
    calc(nums, n, &max, &min, &avg);
    printf("Max = %d\n Min = %d\n Average = %.2f\n",
           max, min, avg);
}
```